

CLAIMS

What is claimed is:

1 1. A microprocessor including:
2 an instruction decoder to decode multiple threads of
3 instructions, the instruction decoder having,
4 an instruction decode pipeline to decode each input
5 instruction associated with each thread; and
6 a shadow pipeline operating in parallel with the
7 instruction decode pipeline, the shadow pipeline
8 including,
9 an instruction thread identification pipeline
10 to associate a thread identification with each
11 instruction being decoded in the instruction decode
12 pipeline, and
13 an instruction valid pipeline to associate a
14 valid indicator with each instruction being decoded
15 in the instruction decode pipeline.

1 2. The microprocessor of claim 1 wherein the instruction
2 decode pipeline and shadow pipeline are physically integrated
3 together into one pipeline within a microprocessor integrated
4 circuit.

1 3. The microprocessor of claim 1 wherein the instruction
2 decoder further has,

3 a pipeline controller coupled to the instruction decode
4 pipeline and the shadow pipeline, the pipeline controller to
5 control in parallel the clocking of each pipestage of the
6 instruction decode pipeline and each pipestage of the shadow
7 pipeline.

1 4. The microprocessor of claim 3 wherein a thread specific
2 clear is received and the pipeline controller invalidates
3 only those instructions in each stalled pipestage of the
4 instruction decode pipeline having a thread identification
5 the same as the thread identification of the thread specific
6 clear.

1 5. The microprocessor of claim 4 wherein the pipeline
2 controller invalidates instructions in each pipestage by
3 setting or clearing the valid indicator in the parallel
4 pipestages of the instruction valid pipeline.

1 6. The microprocessor of claim 3 wherein a thread specific
2 clear is received and the pipeline controller invalidates
3 only those instructions input into each clocked pipestage of

the instruction decode pipeline when a pipestage prior to each clocked pipestage has a thread identification the same as the thread identification of the thread specific clear.

7. The microprocessor of claim 6 wherein the pipeline controller invalidates instructions in each pipestage by setting or clearing the valid indicator in the parallel pipestages of the instruction valid pipeline.

8. The microprocessor of claim 3 wherein a stall associated with a thread identification is received and the pipeline controller determines that the thread identification associated with the stall matches the thread identification of the valid instruction in the next to last pipestage of the instruction decode pipeline and the pipeline controller stalls the next to last pipestage by stopping the clock to the next to last pipestage for the next cycle to hold the valid instruction contained therein.

9. The microprocessor of claim 8 wherein the pipeline controller further determines that a valid instruction is contained within a first pipestage of the instruction decode pipeline other than the next to last pipestage and a valid instruction is contained within a second pipestage

6 immediately subsequent to the first pipestage and the
7 pipeline controller stalls the first pipestage by stopping
8 the clock to the first pipestage for the next cycle to hold
9 the valid instruction contained therein.

1 10. The microprocessor of claim 9 wherein the pipeline
2 controller determines that an instruction is valid by
3 analyzing the valid indicator within the instruction valid
4 pipeline of each pipestage.

1 11. The microprocessor of claim 3 wherein the pipe
2 controller further determines that an invalid instruction is
3 contained within a first pipestage immediately prior to a
4 second pipestage in the instruction decode pipeline and the
5 pipe controller powers down the second pipestage by stopping
6 the clock to the second pipestage until a valid instruction
7 is contained within the first pipestage to conserve power.

1 12. A multithread pipelined instruction decoder comprising:
2 an instruction decode pipeline to decode
3 instructions associated with a plurality of instruction
4 threads; and

5 a shadow pipeline operating in parallel with the
6 instruction decode pipeline, the shadow pipeline
7 including,
8 an instruction thread identification pipeline
9 to associate a thread identification at each pipe
10 stage with each instruction being decoded in the
11 instruction decode pipeline, and
12 an instruction valid pipeline to associated a
13 valid indicator at each pipe stage with each
14 instruction being decoded in the instruction decode
15 pipeline.

1 13. The multithread pipelined instruction decoder of claim
2 12 further comprising:

3 a pipeline controller coupled to the instruction
4 decode pipeline and the shadow pipeline, the pipeline
5 controller to control in parallel the clocking of each
6 pipestage of the instruction decode pipeline and each
7 pipestage of the shadow pipeline.

1 14. The multithread pipelined instruction decoder of claim
2 13 wherein the pipeline controller further controls the
3 invalidation of instructions in each pipestage of the
4 instruction decode pipeline by setting the valid indicator in

5 each pipestage of the instruction valid pipeline to indicate
6 an invalid instruction.

1 15. The multithread pipelined instruction decoder of claim
2 13 wherein the pipeline controller includes powerdown logic
3 to analyze the valid indicator of each pipestage to determine
4 if the next pipestage can be powerdown and to determine if
5 each pipestage should be stalled and the pipeline controller
6 further including clock control logic to determine if a clock
7 signal to each pipestage of the instruction decode pipeline
8 and the shadow pipeline can be stopped to conserve power or
9 preserve data during a stall.

1 16. The multithread pipelined instruction decoder of claim
2 15 wherein the powerdown logic of the pipeline controller to
3 analyze the valid indicator of each pipestage to determine if
4 the next pipestage can be powerdown includes,
5 an inverter for each pipestage to invert the valid
6 indicator to determine if the next pipestage can be powerdown
7 for a next clock cycle.

1 17. The multithread pipelined instruction decoder of claim
2 15 wherein the powerdown logic of the pipeline controller to

3 analyze the valid indicator of each pipestage to determine if
4 each pipestage should be stalled includes,
5 for a determination of a stall condition for the next to
6 last pipestage,
7 an XOR gate to exclusively OR the thread identification
8 of the next to last pipestage with the thread identification
9 of the stall to determined if they match, and
10 a first AND gate to AND the valid indicator of the next
11 to last pipestage with the output from the XOR gate to
12 determine if a pipestage before the next last pipestage
13 should be stalled.

1 18. The multithread pipelined instruction decoder of claim
2 17 wherein the powerdown logic of the pipeline controller to
3 analyze the valid indicator of each pipestage to determine if
4 each pipestage should be stalled includes,

5 for a determination of a stall condition for any other
6 pipestage but the next to last, for each pipestage including

7 a second AND gate to AND the valid indicator of the
8 pipestage for which the determination is being made with the
9 valid indicator of the next pipestage, and

10 a third AND gate to AND the output of the second AND
11 gate with the output from the first AND gate to determine if
12 a given pipestage other than the next to last pipestage
13 should be stalled.

1 19. A method of decoding multiple threads of instructions,
2 comprising:
3 inputting an instruction of a first thread of
4 instructions, a first instruction thread identification, and
5 a first instruction valid indicator into a pipeline in
6 parallel;
7 decoding the instruction of the first thread of
8 instructions;
9 maintaining the parallel association between the
10 instruction of the first thread, the first instruction thread
11 identification, and the first instruction valid indicator
12 during the decoding of the instruction of the first thread of
13 instructions;
14 inputting an instruction of a second thread of
15 instructions, a second instruction thread identification, and
16 a second instruction valid indicator into a pipeline in
17 parallel;
18 decoding the instruction of the second thread of
19 instructions; and
20 maintaining the parallel association between the
21 instruction of the second thread of instructions, the second
22 instruction thread identification, and the second instruction
23 valid indicator during the decoding of the instruction of the
24 second thread of instructions.

1 20. The method of claim 19 further comprising:
2 invalidating only those instructions having a first
3 instruction thread identification in the pipeline when a
4 thread specific clear for the first instruction thread is
5 received.

1 21. The method of claim 20 wherein,
2 the instructions are invalidated by clearing the first
3 instruction valid indicator for each instruction of the first
4 thread of instructions in the pipeline.

1 22. The method of claim 19 further comprising:
2 disabling a clock signal to a next to last pipestage in
3 the pipeline when the instruction valid indicator of the
4 instruction contained within the next to last pipestage
5 indicates a valid instruction and a thread identification of
6 a thread specific stall matches the thread identification of
7 the instruction contained within the next to last pipestage.

1 23. The method of claim 19 further comprising:
2 disabling a clock signal to a pipestage other than the
3 next to last pipestage in the pipeline when the instruction
4 valid indicator of the instruction contained within the

5 pipestage being evaluated indicates a valid instruction and
6 the instruction valid indicator of the instruction of a next
7 pipestage indicates a valid instruction and the next to last
8 pipestage is stalled.

1 24. The method of claim 19 further comprising:
2 enabling a clock signal to a pipestage when the
3 pipestage is not stalled and the pipestage is not powerdown.

1 25. The method of claim 19 further comprising:
2 disabling a clock signal to a pipestage when a prior
3 pipestage contains an invalid instruction as indicated by a
4 valid indicator of the instruction to conserve power.

1 26. A computer including:
2 a memory; and
3 a microprocessor, the microprocessor including,
4 an instruction decoder to decode multiple threads of
5 instructions, the instruction decoder having,
6 an instruction decode pipeline to decode each input
7 instruction associated with each thread; and
8 a shadow pipeline operating in parallel with the
9 instruction decode pipeline, the shadow pipeline
10 including,

11 an instruction thread identification pipeline
12 to associate a thread identification with each
13 instruction being decoded in the instruction decode
14 pipeline, and
15 an instruction valid pipeline to associated a
16 valid indicator with each instruction being decoded
17 in the instruction decode pipeline.

1 27. A method of eliminating invalid instructions within an
2 instruction decoder comprising:
3 receiving a thread specific clear instruction indicating
4 a thread ID of instructions to be cleared from a pipeline;
5 comparing a thread identifier of each instruction within
6 each pipestage of the pipeline to determine if it matches the
7 thread ID of instructions to be cleared from the pipeline;
8 invalidating a valid bit for each instruction having a
9 thread ID matching the thread ID of instructions to be
10 cleared from the pipeline.
11

1 28. The method of claim 17 further comprising:
2 clocking each pipestage of the pipeline to continue to
3 decode the valid instructions within the pipeline as
4 indicated by their valid bits.

1 29. A microprocessor including:

2 an instruction decoder to decode multiple threads of
3 instructions, the instruction decoder having,

4 an instruction decode pipeline,

5 the instruction decode pipeline to decode
6 each input instruction associated with each
7 thread, and

8 the instruction decode pipeline to
9 maintain a thread identification and a valid
10 indicator in parallel with each instruction
11 being decoded in the instruction decode
12 pipeline.

1 30. The microprocessor of claim 29 wherein the instruction
2 decode pipeline includes a series of registers within a
3 microprocessor integrated circuit.

1 31. The microprocessor of claim 29 wherein the instruction
2 decoder further has,

3 a pipeline controller coupled to the instruction decode
4 pipeline, the pipeline controller to control the clocking of
5 each pipestage of the instruction decode pipeline.

1 32. The microprocessor of claim 31 wherein a thread specific
2 clear is received and the pipeline controller invalidates
3 only those instructions in each stalled pipestage of the
4 instruction decode pipeline having a thread identification
5 the same as the thread identification of the thread specific
6 clear.

1 33. The microprocessor of claim 31 wherein a thread specific
2 clear is received and the pipeline controller invalidates
3 only those instructions input into each clocked pipestage of
4 the instruction decode pipeline when a pipestage prior to
5 each clocked pipestage has a thread identification the same
6 as the thread identification of the thread specific clear.

1 34. The microprocessor of claim 31 wherein a stall
2 associated with a thread identification is received and the
3 pipeline controller determines that the thread identification
4 associated with the stall matches the thread identification
5 of the valid instruction in the next to last pipestage of the
6 instruction decode pipeline and the pipeline controller
7 stalls the next to last pipestage by stopping the clock to
8 the next to last pipestage for the next cycle to hold the
9 valid instruction contained therein.

1 35. The microprocessor of claim 38 wherein the pipeline
2 controller further determines that a valid instruction is
3 contained within a first pipestage of the instruction decode
4 pipeline other than the next to last pipestage and a valid
5 instruction is contained within a second pipestage
6 immediately subsequent to the first pipestage and the
7 pipeline controller stalls the first pipestage by stopping
8 the clock to the first pipestage for the next cycle to hold
9 the valid instruction contained therein.

1 36. The microprocessor of claim 31 wherein the pipe
2 controller further determines that an invalid instruction is
3 contained within a first pipestage immediately prior to a
4 second pipestage in the instruction decode pipeline and the
5 pipe controller powers down the second pipestage by stopping
6 the clock to the second pipestage until a valid instruction
7 is contained within the first pipestage to conserve power.